

FIG. 1

FIG. 2A

```

1  <!-- BEGIN Event Tracking Code -->
2  <script language="javascript1.2">
3  function recordit(k, w)
4  //This function records the link. Parameters k and w are passed. K is the link and w is an array of stored
5  functions.
6  {
7      var gettext="NA";
8      //This initializes the variable used to hold the text that the link wrapped.
9      var iloc=0;
10     while (document.links[iloc]!=k) {
11         iloc++;
12     }
13     //This is a loop which determines which element in the array the link that was passed.
14     if (w[iloc]){w[iloc]();}
15     //This checks to see if a function was stored for that link. If so, it executes the stored function.
16     document.all?gettext=k.innerHTML:gettext=k.text;
17     //This sets the gettext variable to the text or html that was wrapped by the link.
18     document.images.hitboximage.src="http://www3.hitbox.com/logs/log.cgi?link="+k.href+"&iloc="+iloc+"
19     &named="+gettext;
20     //This sends the link information back to the hitbox servers.
21 }
22

```

FIG. 2B

```

23 function initial()
24 //This function sets the onclick event for each link and stores prior events associated with it, if any.
25 {
26     var storef = Array(document.links.length);
27     //This array will store prior functions.
28     for (var i = 0; i < document.links.length; i++) {
29         //This loop runs through all the documents on a page.
30         if (document.links[i].onclick) storef[i]=document.links[i].onclick;}
31         //If a function is associated with that link it is stored in an array.
32         document.links[i].onclick = function () {
33             recordit(this, storef);
34             return true;
35             //Here is where the onclick event for each link is associated with the recordit function.
36         }
37     }
38 }
39
40 window.onload = initial;
41 //This starts the process when the window has loaded.
42 </script>
43
44 <IMG SRC="http://stats.hitbox.com/buttons/ct0.gif" HEIGHT=68 WIDTH=82 BORDER=0
45 NAME="hitboximage">
46 <!-- This is the initial image used to record a page view.
47

```

FIG. 3

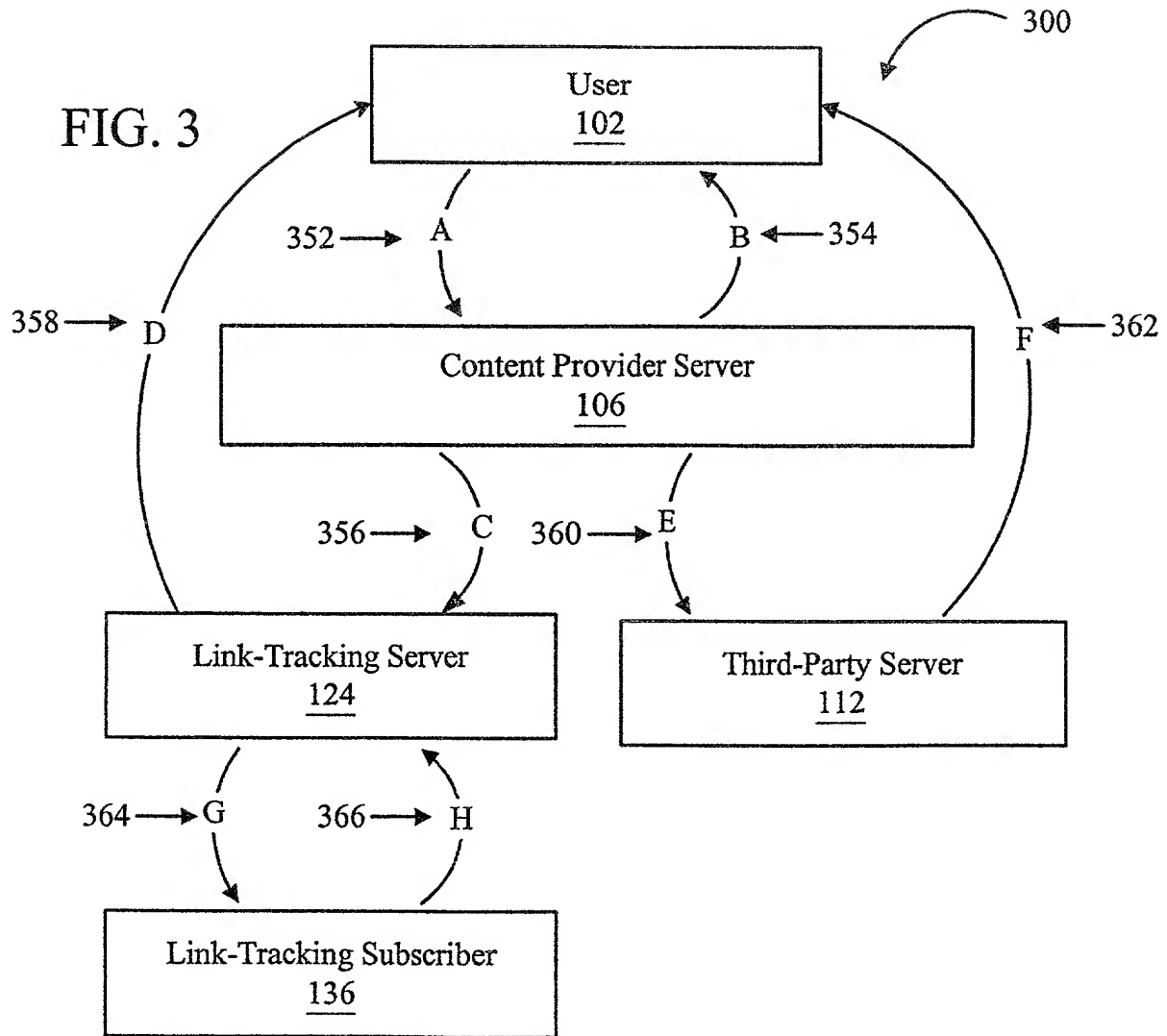


FIG. 4A

```

1  <script language="javascript1.2">
2  This code demonstrates how to record downloads and exit links using the invention.
3  _hbi=new Image();
4  _ot=0;
5  //Initializes variables.
6
7  function _gf(_x){
8      //This function retrieves the file or document name requested in a URL.
9      _ed = (_x.length);
10     _be = ((_x.lastIndexOf("/") + 1));
11     _fn = (_x.substring(_be, _ed));
12     return _fn;
13 }
14
15 function _sb(_k){
16     //This function checks to see if the link is an exit link.
17     eval("var _gt=/(\w*\.\.)*(\w*\.\.)/i");
18     _gt.exec(location.hostname);
19     //This parses the link for domain name.
20     var _fr="co.com.org.net.";
21     if(_fr.indexOf(RegExp.$2)!=-1){
22         eval("var _uf=/(\w*\.\.)*(\w*\.\.)(\w*\.\.)/");
23         _uf.exec(location.hostname);
24     }
25     //This is additional checking in case the link is to a foreign domain.
26     if(_k.href.indexOf(RegExp.$2)===-1){ _hbi.src="http://ehg.hitbox.com/HG?el="+escape(_k.href); }
27     //If the domain name is not the same as the current document, the information is sent back to the hitbox
28     servers.
29     return true;
30 }

```

FIG. 4B

```

function _ri(_k,_w){
//This function receives the link and checks if it is a download. Parameters _k and _w are passed. _k is the
link and _w is an array of stored functions.
    var _ilc=0;
    while(document.links[_ilc]!=$_k){_ilc++;}
    if(_w[_ilc]){_w[_ilc]();}
    //This is a loop which determines which element in the array the link that was passed.
    _fl=_gf(_k.pathname);
    //Sets variable _fl to the file or document name in the link.
    if(_fl!=""){
        //Checks if name exists
        var _pt=new RegExp("\.?html?\.asp\.cfm\.jsp\.cgi\.php[3-4]?\.pl\.taf\.dll"."i");
        if(!_pt.test(_fl)){
            //Checks to see if file extension is among a list of known HTML-types.
            _hbi.src="http://ehg.hitbox.com/HG?fn="+escape(_fl);}
            //If the file is not among that list, the file information is sent back to the hitbox servers.
        else{
            _sb(_k);}
        //If info was not sent back to hitbox server, it checks for exit links.
    else{
        _sb(_k);
        //If info was not sent back to hitbox server, it checks for exit links.
    }
    return true;
}

```

FIG. 4C

```

56 function _it() {
57 //This function sets the onclick event for each link and stores prior events associated with it, if any.
58
59     if(_ot==1)_wo();
60     //If an onload event was previously set, executes it now.
61     var _sf=new Array(document.links.length);
62     //This array will store prior functions.
63     for(var i=0;i<document.links.length;i++){
64         _rv="true";
65         if(document.links[i].onclick) {
66             //This loop runs through all the documents on a page.
67             _sf[i]=document.links[i].onclick;
68             //If a function is associated with that link it is stored in an array.
69             if(_sf[i].toString().indexOf("return false")!=-1) _rv="false";
70             //If the return value of the stored function is false, the return value for the new function is
71             set to false.
72         }
73         eval("document.links[i].onclick=function() { _ri(this,_sf);return "+_rv+"; }");
74         //Here is where the onclick event for each link is associated with the _ri function.
75     }
76
77     if(window.onload){_wo=window.onload;_ot=1;};window.onload=_it;
78     //If a function is already associated with the window loading, it is stored and executed later.
79 }
</script>

```